INTERNATIONAL ORGANISATION FOR STANDARDISATION

ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC1/SC29/WG11

CODING OF MOVING PICTURES AND ASSOCIATED AUDIO INFORMATION

ISO/IEC JTC1/SC29/WG11
MPEG96/1047
July 1996

Source: AT&T and Sharp Corporation

Status: Proposal

Title: MPEG-4 Video VM Syntax and Semantics (including Scalability)

Authors: A. Puri, R. L. Schmidt, B. G. Haskell, H. Katata, T. Aono, N. Ito

1. Introduction

To remove inconsistencies, reduce overhead and enable important functionalities, a revision of the syntax and semantics of the current MPEG-4 Video Verification Model (VM) is proposed. The revised video syntax included in this document supports all of the current features in VM2.2, and in addition, also enables functionalities such as scalability and provides flexibilities that may be useful for error resilience and multi-viewpoint coding. The proposed syntax consists of the following class hierarchy:

- VideoSession (VS)
- VideoObject (VO)
- VideoObjectLayer (VOL)
- VideoObjectPlane (VOP)

Within the context of video experiments, it can be said that a VS is a collection of one or more VO's, a VO can consist of one (nonscalable) or more layers (scalability) and that each layer consists of an ordered sequence of snapshots in time called VOPs. Thus there can be several VO's (VO0, VO1,..) in a VS and for each VO, there can be several scalability layers (VOL0, VOL1,..) and each scalability layer consists of time sequence of VOPs (VOP0, VOP1,..), which are basically snapshots in time. A VO can be of arbitrary shape (rectangular is a special case). For nonscalable coding only one VOL (VOL0) exists per VO. In scalable coding VOL0 would be the base layer and VOL1 the first enhancement layer and so forth. Figure 1 shows the hierarchical structure of the proposed syntax.

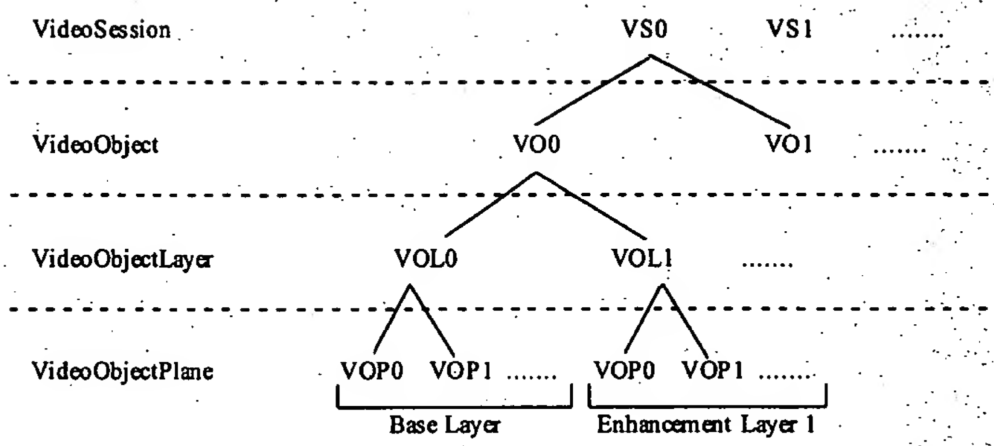

Figure 1 : Hierarchy in the proposed video syntax

In Section 2 we provide the complete syntax and semantics including that for generalized scalability which is based on MPEG-2 Temporal Scalability syntax and is extended to provide Object-based Temporal Scalability. In Section 3 we provide a description of the generalized scalability. Section 4 provides a summary of this document.

2. Syntax and Semantics

2.1 Video Session

Syntax	No. of bits	Mnemonic
VideoSession() {		
video_session_start_code	32	
do {		
do {		
VideoObject()		
} while (nextbits() == video_object_start_code)		
if(nextbits() != session_end_code)		
video_session_start_code	32	
} while (nextbits() != video_session_end_code)		
video_session_end_code	32	
}		

2.2 Video Object

Syntax	No. of bits	Mnemonic
VideoObject() {		
video_object_start_code	24+3	
object_id	5	
do {		
VideoObjectLayer()		
} while (nextbits() == video_object_layer_start_code)		
next_start_code()		
}		

object_id
It uniquely identifies a layer. It is a 5-bit quantity with values from 0 to 31.

2.3 Video Object Layer

Syntax	No. of bits	Mnemonic

```
VideoObjectLayer() {
  video_object_layer_start_code                                   28
  layer_id                                                         4
  layer_width                                                     10
  layer_height                                                    10
  quant_type_sel                                                   1
  if (quant_type_sel) {
    load_intra_quant_mat                                           1
    if (load_intra_quant_mat)
      intra_quant_mat[64]                                       8*64
    load_nonintra_quant_mat                                        1
    if (load_nonintra_quant_mat)
      nonintra_quant_mat[64]                                    8*64
  }
  intra_dcpred_disable                                             1
  scalability                                                      1
  if (scalability) {
    ref_layer_id                                                   4
    ref_layer_sampling_direc                                       1
    hor_sampling_factor_n                                          5
    hor_sampling_factor_m                                          5
    vert_sampling_factor_n                                         5
    vert_sampling_factor_m                                         5
    enhancement_type                                               1        SHARP
  }
  do {
    VideoObjectPlane()
  } while (nextbits() == video_object_plane_start_code}
  next_start_code()
}
```

layer_id
It uniquely identifies a layer. It is a 4-bit quantity with values from 0 to 15. A value of 0 identifies the first independently coded layer.

layer_width, layer_height
These values define the spatial resolution of a layer in pixels units.

scalability
This is a 1-bit flag which indicates if scalability is used for coding of the current layer.

ref_layer_id
It uniquely identifies a decoded layer to be used as a reference for predictions in the case of scalability. It is a 4-bit quantity with values from 0 to 15.

ref_layer_sampling_direc
This is a 1-bit flag whose value when "0" indicates that the reference layer specified by ref_layer_id has the same or lower resolution as the layer being coded. Alternatively, a value of "1" indicates that the resolution of reference layer is higher than the resolution of layer being coded resolution.

hor_sampling_factor_n, hor_sampling_factor_m
These are 5-bit quantities in range 1 to 31 whose ratio hor_sampling_factor_n/hor_sampling_factor_m indicates the resampling needed in horizontal direction; the direction of sampling is indicated by ref_layer_sampling_direc.

vert_sampling_factor_n, vert_sampling_factor_m
These are 5-bit quantities in range of 1 to 31 whose ratio vert_sampling_factor_n/vert_sampling_factor_m indicates the resampling needed in vertical direction; the direction of sampling is indicated by ref_layer_sampling_direc.

SHARP

enhancement_type
This is a 1-bit flag which indicates the type of an enhancement structure in a scalability. It has a value of "1" when an enhancement layer enhances a partial region of the base layer. It has a value of "0" when an enhancement layer enhances entire region of the base layer. The default value of this flag is "0".

Other syntax elements such as quant_type_sel and intro_dcpred_disable in the Video Object Layer have the same meaning described in VM.

2.4 Video Object Plane

Syntax	No. of bits	Mnemonic
VideoObjectPlane() {		
video_object_plane_start_code	32	
plane_temp_ref	16	
plane_visibility	1	
plane_of_arbitrary_shape	1	
if (plane_of_arbitrary_shape) {		
plane_width	10	
plane_height	10	
if (plane_visibility) {		
plane_composition_order	5	
plane_hor_spatial_ref	10	
marker_bit	1	
plane_vert_spatial_ref	10	
plane_scaling	3	
if (scalability && enhancement_type) SHARP		
background_composition	1	
}		
shape()		
}		
plane_coding_type	2	
if (plane_coding_type == 1 || plane_coding_type == 2) {		
plane_fcode_for	2	
if (plane_coding_type == 2) {		
plane_fcode_back	2	
plane_dbquant	2	
}		
else {		
plane_quant	5	
}		
if (!scalability) {		
separate_motion_texture	1	
if (!separate_motion_texture)		
combined_motion_texture_coding()		
else {		
motion_coding()		
texture_coding()		
}		
}		
else {		
if (background_composition) {		
load_backward_shape	1	
if (load_backward_shape) {		
backward_shape()		
load_forward_shape	1	
if (load_forward_shape)		

SHARP

```
      forward_shape()
    }
  }
  ref_select_code                                                     2
  if(plane_coding_type == 1 || plane_coding_type == 2) {
    forward_temporal_ref                                              10
    if(plane_coding_type == 2) {
      marker_bit                                                      1
      backward_temporal_ref                                           10
    }
  }
  combined_motion_texture_coding()
 }
}
```

background_composition
This flag only occurs when scalability flag has a value of "1". The default value of this flag is "0". This flag is used in conjunction with enhancement_type flag. If enhancement_type is "1" and this flag is "1", background composition is performed. If enhancement type is "1" and this flag is "0", background is repeated from the nearest frame in base layer. Further, if enhancement type is "0" no action needs to be taken as a consequence of any value of this flag.

shape()
The *shape()* function generates the format of the coded data of a current shape (alpha plane).

Syntax	No. of bits	Mnemonic
shape() {		
binary_shape	1	
if(binary_shape) {		
do {		
first_QT_code	1-2	
if(first_QT_code=="00")		
subsequent_QT_codes		
} while (count of macroblock != total number of macroblocks)		
} else{		
do {		
first_QT_code		
if(first_QT_code=="00") {		
subsequent_QT_codes		
VQ_codes	0-128	
}		
} while (count of macroblock != total number of macroblocks)		
}		
}		

load_backward_shape
If this flag is "1", backward_shape of the previous VOP is copied to forward_shape for the current VOP and backward_shape for the current VOP is decoded from the bitstream. If not, forward_shape for the previous VOP is copied to forward_shape for the current VOP and backward_shape for the previous VOP is copied to backward_shape for the current VOP.

backward_shape()
It specifies the format of coded data for backward_shape and is identical to that of *shape()*. A boundary rectangle of *backward_shape()* is same as the entire image.

load_forward_shape
This flag is "1" if forward_shape will be decoded from a bitstream.

SHARP

SHARP

forward_shape()
It specifies the format of coded data for forward_shape and is identical to that of *shape()*. A boundary rectangle of *forward_shape()* is same as the entire image.

ref_select_code
This is a 2-bit code which indicates prediction reference choices for P- and B-VOPs in the enhancement layer with respect to decoded reference layer identified by ref_layer_id.

forward_temporal_ref
An unsigned integer value which indicates temporal reference of the decoded reference layer VOP to be used for forward prediction (Table 1 and Table 2)

backward_temporal_ref
An unsigned integer value which indicates temporal reference of the decoded reference layer VOP to be used for backward prediction (Table 2).

3. Generalized Scalability

Generalized scalability involves more than one layer in VideoObjectLayer. Considering the case of two layers, a lower layer and an enhancement layer, the spatial resolution of each layer may be either the same or different; when the layers have different spatial resolution, (up or down) sampling of lower layer with respect to the enhancement layer becomes necessary for generating predictions. If the lower layer and the enhancement layer are temporally offset, irrespective of the spatial resolutions, motion compensated prediction may be used between layers. When the layers are coincident in time but at different resolution, motion compensation may be switched off to reduce overhead.

The reference VOPs for prediction are selected by reference_select_code as described in Tables 1 and 2. In coding P-VOPs belonging to an enhancement layer, the forward reference can be one of the following three: the most recent decoded VOP of enhancement layer, the most recent VOP of the lower layer in display order, or the next VOP of the lower layer in display order.

In B-VOPs, the forward reference can be one of the two: the most recent decoded enhancement VOP or the most recent lower layer VOP in display order. The backward reference can be one of the three: the temporally coincident VOP in the lower layer, the most recent lower layer VOP in display order, or the next lower layer VOP in display order.

Table 1 : Prediction reference choices for P-VOPs in the object-based temporal scalability.

ref_select_code	forward prediction reference
00	Most recent decoded enhancement VOP belonging to the same layer.
01	Most recent VOP in display order belonging to the reference layer.
10	Next VOP in display order belonging to the reference layer.
11	Temporally coincident VOP in the reference layer (no motion vectors)

Table 2 : Prediction reference choices for B-VOPs in the case of scalability.

ref_select_code	forward temporal reference	backward temporal reference
00	Most recent decoded enhancement VOP of the same layer	Temporally coincident VOP in the reference layer (no motion vectors)
01	Most recent decoded enhancement VOP of the same layer.	Most recent VOP in display order belonging to the reference layer.

10	Most recent decoded enhancement VOP of the same layer.	Next VOP in display order belonging to the reference layer.
11	Most recent VOP in display order belonging to the reference layer.	Next VOP in display order belonging to the reference layer.

The enhancement layer can contain I, P or B-VOPs, but the B-VOPs in the enhancement layer behave more like P-VOPs at least in the sense that a decoded B-VOP can be used to predict the following P or B-VOPs.

When the most recent VOP in the lower layer is used as reference, this includes the VOP that is temporally coincident with the VOP in the enhancement layer. However, this necessitates use of lower layer for motion compensation which requires motion vectors.

If the coincident VOP in the lower layer is used explicitly as reference, no motion vectors are sent and this mode can be used to provide spatial scalability. Spatial scalability in MPEG-2 uses spatio-temporal prediction, which is accomplished here by using the prediction modes available for B-VOPs.

Since the VOPs can have a rectangular shape (picture) or an irregular shape, both the traditional as well as object based temporal and spatial scalabilities become possible.

We explain next the meaning of enhancement_type flag in more detail. As an example, Figure 2 shows an entire image containing several types of regions for example a road, a car, and mountains. Both the base layer with enhancement_type being "0" and the base layer with enhancement_type being "1" are coded with lower picture quality which means that either the frame rate is lower or the spatial resolution is lower. At the enhancement layer of the scalability, enhancement_type flag distinguishes the following two cases.

- When this flag is "1", the enhancement layer increases the picture quality of a partial region of the base layer. For example, in Figure 2, VO0 is an entire frame and VO1 is the car in the frame. The temporal resolution or the spatial resolution of the car is enhanced.
- When this flag is "0", the enhancement layer increases the picture quality of the entire region of the base layer. For example, in Figure 2, if VO0 represents an entire frame, VO1 is also the entire frame. Then the temporal or spatial resolution of entire frame is enhanced. If VO0 represents the car, VO1 is also the car which is enhanced in terms of temporal or spatial resolution.

4. Summary

A new syntax and clear semantics for are proposed. The syntax class hierarchy consists of the following:

- VideoSession (VS)
- VideoObject (VO)
- VideoObjectLayer (VOL)
- VideoObjectPlane (VOP)

This syntax not only supports all features of the current VM but also important functionalities such as object based scalability. For nonscalable coding, the overhead is reduced by moving the parameters that do not change from a VOP to the level of VOL which occurs less frequently. It introduces scalability in a structured manner. Since the proposed scalability syntax is based on the simplification of MPEG-2 scalability syntax with minimal extensions necessary to enable object scalability it is efficient. In addition to scalability the flexibilities offered by the syntax are expected to be useful for error resilience and multi-viewpoint functionalities.

In addition, issues in generalized scalability including how predictions are formed are explained in detail. Traditional spatial and temporal scalabilities suitable for the lower bitrates MPEG-4 is addressing are derived as a subset of the generalized scalability syntax. Scalability on arbitrary shaped objects as well as rectangular (picture) objects is also supported by the generalized scalability.

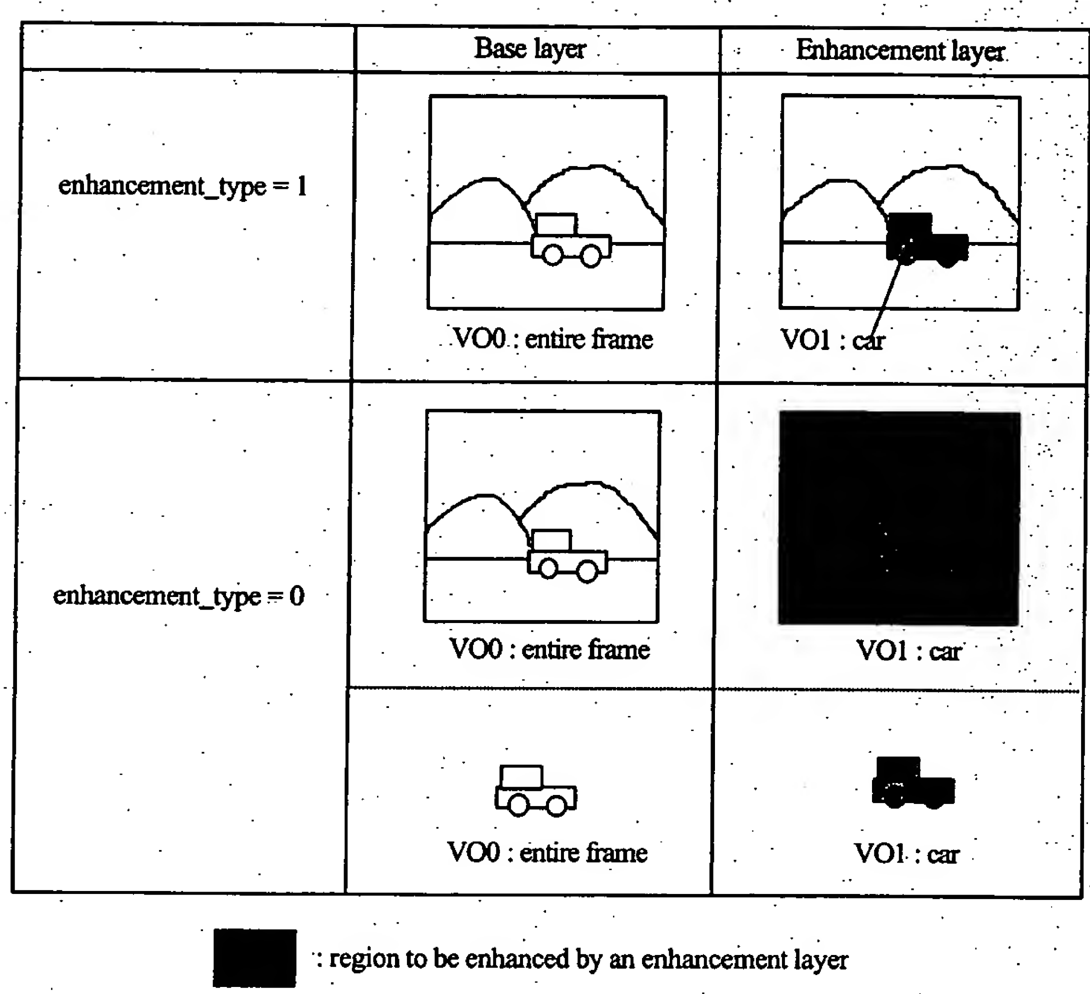

Figure 2 : Example of a region to be enhanced.